IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received September 26, 2018, accepted October 15, 2018, date of publication November 12, 2018,
date of current version December 31, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2877452

Deep Fusion Feature Learning Network for MI-EEG Classification

JUN YANG[1,2], SHAOWEN YAO[1], AND JIN WANG[1]
[1]School of Information Science and Engineering, Yunnan University, Kunming 650504, China
[2]School of Information Science and Automation, Kunming Science and Technology University, Kunming 650504, China

Corresponding author: Shaowen Yao (paradisewolf@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61863036 and in part by the Natural Science Foundation of China under Grant 61640306.

ABSTRACT Brain–computer interfaces (BCIs) are used to provide a direct communication between the human brain and the external devices, such as wheelchairs and intelligent apparatus, by interpreting the electroencephalograph (EEG) signals. Recently, motor imagery EEG (MI-EEG) has become an active research field where a subject's active intent can be detected. The accurate decoding of MI-EEG signals is essential for effective BCI systems but also very challenging due to the lack of informative correlation between the signals and the brain activities. To improve the precision performance of a BCI system, accurate feature discrimination from input signals and proper classification are necessary. However, the traditional deep learning scheme is failed to generate spatio-temporal representation simultaneously and capture the dynamic correlation for an MI-EEG sequence. To address this problem, we propose a long short-term memory network combined with a spatial convolutional network that concurrently learns spatial information and temporal correlations from raw MI-EEG signals. In addition, spectral representations of EEG signals are obtained via a discrete wavelet transformation decomposition. In order to achieve even higher learning rates and less demanding initialization, we employ a batch normalization method before training and recognition. Various experiments have been performed to evaluate the performance of the proposed deep learning architectures. Results indicate a high level of accuracy over both the public data set and the local data set. Our method can also serve as a useful and robust model for multi-task classification and subject-independent movement class decoder across many different methods.

INDEX TERMS Motor imagery electroencephalograph (MI-EEG) brain computer interfaces (BCI), long short-term memory (LSTM), convolutional neural networks (CNN).

I. INTRODUCTION

Brain computer Interfaces (BCI) [1]–[3] play an essential role as information pathways between the human brain and external world [4] when the peripheral nerve pathway is severely damaged by disease such as apoplexy or degenerative pathologies. Among the different types of BCI mechanisms, motor imagery Electroencephalography (MI-EEG) [5], [6], is considered to be the most flexible method since it has been proved promising in discriminating different brain activities. Motor imagery (MI) is a mental process [7] with which a subject can encode its intentions in EEG by imagining performing a certain action such as lifting right hand or moving feet. Actually, when people imagine movement of unilateral limb, the wave power of μ (8-12Hz EEG) and β (18-26hz EEG) rhythm [8] from the contralateral motor sensory cortex decrease while the wave power increase in the ipsilateral. These correlative phenomena [9] are named event related desynchronization (ERD) and event related synchronization (ERS) respectively which could be experimentally observed through various brain activity measuring techniques. The most popular technologies to record such brain signals is EEG. For MI-EEG, signals inherently lack of sufficient spatial resolution and activity insights of deep brain structures. the EEG sequence usually have low signal-to-noise ratio (SNR) and contains a large amount of useless information [10], which makes it very challenging to accurately understand brain dynamics and classify different motor imageries. Thus, the key issue concerning an EEG-based BCI system is to accurately interpret EEG signals from user's intent. At present, four state-of-the-art methods are mainly applied in BCIs which are based on MI-EEG. Linear Discriminant Analysis (LDA) [11] is a machine learning algorithm that attempts to represent one dependent variable as a linear combination of other features. It performs well in linear application analysis but is not much accurate at nonlinear BCI systems. Support Vector Machine (SVM) [12] can generate a non-linear decision boundary by projecting the data through a non-linear function to a high dimensional space. It provides good nonlinear mapping ability and generalization capacity. Naive Bayes (NB) [13] is based on Bayes' theorem with independence assumptions where the presence of a class feature is unrelated to rest. It cannot handle very noisy EEG data. Deep neural networks (DNN) are based on the Back Propagation (BP) algorithm and present strong nonlinear fitting capability [14]. Various previous studies apply the developed deep neural networks such as convolutional neural networks (CNN) and recurrent neural networks (RNN) for decoding MI-EEG.

CNN are widely applied in MI-EEG recognition due its ability of extracting the most discriminant features (high-level features) for classification [15]. Various works utilizing deep convolutional neural network focus on obtaining discriminative features from input signal, they do not consider disentangle factors and variation parameters [16]. The discrete wavelet transforms (DWT), as a new development of Fourier transform, can express the feature information of a sequence both in temporal and spectral domain [17] Taking this consideration, we employ the CNN and DWT to help us locate the most informative channels and explore the time-frequency feature from the MI-EEG data.

Recurrent neural networks (RNNs) [18] is a class of artificial neural networks with recurrent connections able to model sequential data and exhibit dynamic temporal behavior for sequence recognition and prediction. RNNs consist of high dimensional hidden states with non-linear characteristic [19]. The hidden states work as the memory of the architecture and current states of the hidden layers are correlated with their previous ones. The motivation behind using RNNs is their ability to exploit sequential information since their structure allows remembering and processing past complex signals for long time periods. For each timestep, RNNs can map an input signal to the output signal and predict the sequence in the next timestep. While RNNs can make use of information in arbitrarily long sequences, they are actually limited to only few steps back and are weak in vanishing and exploding gradient problems. Long short-term memory (LSTM) networks [20], [21], as a novel class of RNNs, are much more efficient at capturing long-term dependencies, thus, they are popular and effective in reducing the effects of vanishing and exploding gradients when training traditional RNNs. By this method the composition of hidden units is changed from "sigmoid" or "tanh" to memory cells, for controlling the inputs and outputs applied to the gates. These gates control the information flow in hidden neurons and preserve the extracted features from previous timesteps. Compared with the RNN, LSTM can "look back" further and reach less risk on vanishing gradient and over-fitting [22].

For traditional neural network methods, the initial weights need be chosen with experience and patience, which is one major obstacle of their widespread application. In addition, the traditional machine learning and neural network methods neglect the significance of feature extraction and selection. The signal is prone to transform using independent component analysis (ICA) or common spatial patterns (CSP) [23] to obtain spatial components. Furthermore, these deep neural network recognitions [24] also neglect the long-range dynamical correlation which is vital to the event related potential (ERP) from MI-EEG [25]. On the other hand, the training of deep neural network is time-consuming and Batch Normalization (BN) is thus used to normalize the input before the training. BN is technique that normalizes activations in intermediate layers with zero mean and unit variance [26].

In brief, a novel convolutional neural networks (CNN) were used to find out the most-informative linear subspace of the original channels. Then, DWT is designed to capture the temporal and spectral features and BN is used for training smoothly. Finally, a special kind of recurrent neural network (RNN) called long short-term memory (LSTM) is developed as regression algorithm to capture the temporal dynamics and recognized our MI-EEG. Our method is different from the aforementioned methods in designing convolution layers and DWT to select the spatial and time-frequency attributes for followed LSTM recognition. This work analyzes each channel of MI-EEG and extract its effective wavelet coefficients. The LSTM is meanwhile used as the regression algorithm to capture the temporal dynamics of the signal and help us capture the long-range sequence feature information.

In this study, we propose the combination of convolutional neural networks (CNN) and long short-term memory (LSTM) to recognize our MI-EEG. In particular, DWT is designed to model temporal and spectral features and BN is used for smooth training. In brief, the primary contributions of this study are listed as follows:

1) We present a deep joint network that leverages CNN and DWT to capture high-level spatial feature representations and time-frequency information from raw MI-EEG signals for recognition, respectively.
2) We employ Batch Normalization for faster training and for reducing the sensitivity to initialization before the LSTM regression algorithm, which captures the temporal dynamics and the state transitions after feature extraction.
3) We extensively evaluate our model using public and private datasets for demonstration the efficacy and practicality of our approach. The experimental results illustrate that the proposed model can achieve high levels of accuracy over both the public (87.36%) and the local datasets (86.71%).

The rest of the paper is as follows. In Section II, the experiment datasets and data processing are presented in Section III, we describe the proposed deep learning archi-

TABLE 1. Properties of raw materials.

Datasets	Private	Public		
	D_1	D_2	D_3	D_4
subject	6	7	12	5
electrode	64	64	32	64
Sample Rate (Hz)	500	1000	10000	500
Class of task	left hand, right foot	left, right hand	left hand, tongue	left, right hand

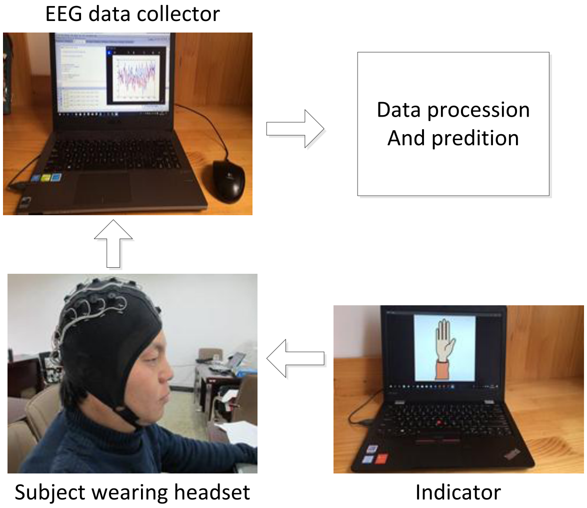

FIGURE 1. EEG collection.

tecture for feature extracting and classification. Finally, various experiments and visualizations validate effectiveness and robustness of the proposed architecture.

II. DATASET AND PRE-PROCESSING

Before EEG decoding and interpretation, EEG feature representation and the EEG preprocessing should be elaborated.

A. DATA ACQUISITION AND REPRESENTATION

The proposed scheme has been evaluated on both public data and private data collected in our lab. The MI-EEG data sequence was randomly separated into labelled training subset (almost 70% positive and negative class samples of total experiment data sets) and testing subset (almost 30% of the rest). The data details are shown in Table 1. In this study, all data is labeled with two categories. D1 is the private dataset collected during our own experiments.

In the experiment, six healthy subjects with mean age of 27.8 years were asked to wear the EEG device and sit in front of a computer screen which provides guidance then performing certain imaginary actions (see Fig. 1). Each trial of the collected dataset includes four sessions of motor imagery process, among which the first two sessions are recorded without feedback (left hand and right foot imagery) and the other sessions have incorporated online feedback. D2, D3 and D4 are public datasets from the BCI competition III and IV data sets. Concretely, our experiment input is sequence data which denoted as 3-D shape as (time, 1, channel). The time length is calculated by sample rate and trial time such as when we use the sampling rate of 500Hz for a total time of 10s for a subject, the calculation of length is 5000. The order of the electrodes (channel) which was arranged and stored in our experiment consistent with the public data.

B. DATA PRE-PROCESSING

Before feature extraction, recognition and prediction are introduced to the EEG sequence, several pre-processing operations were necessary, as stated below:

1) **Referencing**[27]. In this experiment, the vertex electrode (C_z) is used as reference electrode. We define the M × N data matrix which contains the two-dimension information of electrodes and the EEG recordings are referenced to C_z by subtraction. This process can be performed by the following transformation:

$$C_z(V_m) = (I - V_{C_z})V_m \tag{1}$$

where I is the $M \times M$ identity matrix.

$$V_{C_z} = \begin{bmatrix} 0 & \cdots & 1 & \cdots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \cdots & 1 & \cdots & 0 \end{bmatrix} \tag{2}$$

It is an $M \times M$ matrix with ones in the C_z electrode corresponding column. V_m is another electrode sequence data which need to be referenced.

2) **Electrode Selection**. The main correlated locations of the 9 MI-EEG electrodes are P_3, P_4, C_3, C_4, O_1, O_2, P_z, F_z and C_z. The reference electrode C_z was placed on the central top of the skull [28].

3) **Artifact removal**. The recorded EEG data included some other irrelevant signals such as electrocardiograms, myoelectricity and electro-oculogram, which bring great difficulties to the signal analysis and processing. To address this challenge, independent component analysis (ICA) and principal component analysis (PCA) were frequently used. The PCA method mainly involve that unwanted artifacts are removed and pure signal is preserved after the raw EEG is decomposed into independent components. The ICA however employ model to estimate signal source and make nonlinear transformation which avoid lose key information against decomposition. In this work, we choose the ICA as our artifact removal approach.

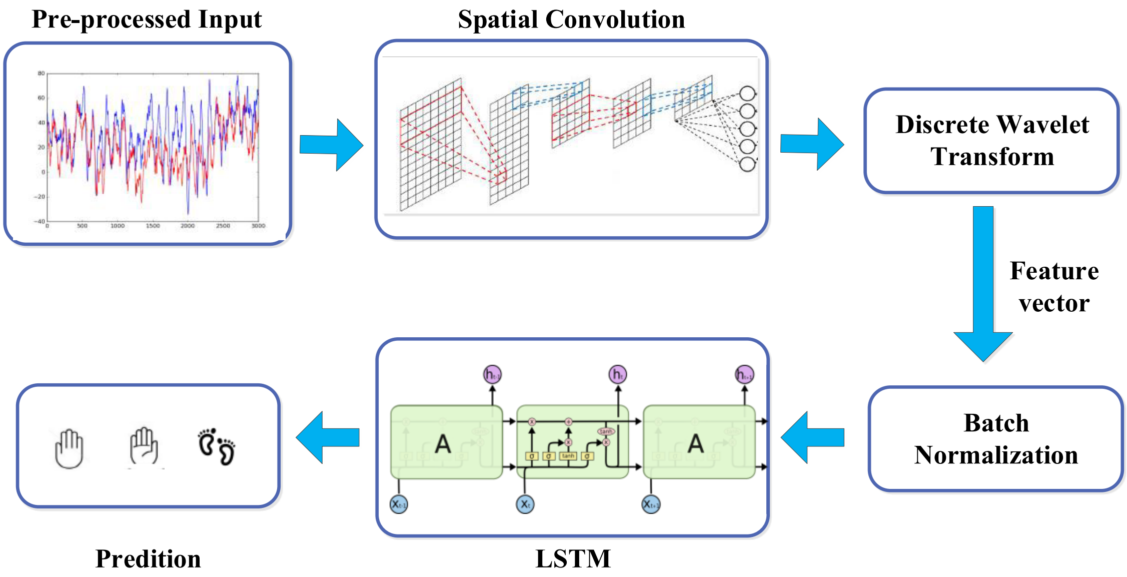

FIGURE 2. The decomposition of proposed deep structure.

4) **Signal filtering**. The principal component of MI-EEG was distributed in the frequency band of the μ and β rhythm mentioned above. We used the bandpass filter to process our data.

III. SPATIAL CONVOLUTION

The proposed architecture is inspired by the typical MI-EEG feature extraction pipeline which consists of spatial convolution transformation and time-frequency analysis, as shown in Fig.2. Our deep architecture combined a spatial convolutional layer, a hierarchical feature extractor with DWT, and a LSTM that is able to process sequential data and capture temporal dynamics in the neural data.

A. THE SPATIAL CONVOLUTION NEURAL NETWORK

The purpose of applying the CNN method was to define the most informative or task-modulated linear subspace of the original channels. The weights of the model are updated through the algorithm of error backpropagation. The convolutional layer performed spatial filtering on the input EEG signal. Various parameters can be optimized for the CNN network which leads to many possible configurations. The major parameters for a CNN network design are show in Table 2. Parameters can be first initialized in simple ways and then weights are trained using back-propagation. We found that appropriate parameter initialization can significantly reduce the chance of overfitting. Thus, we actually choose a dropout probability of 0.5 to counter overfitting. The input signal was a vector with a shape of $(N\ 1, C)$. N and C are the samples and channel numbers of input for each subject, respectively. The 16 spatial features are learned through the CNN structures. If CNN have high classification accuracy, these spatial features which we denote as S has the direct linear map relationship with the EEG sample label.

TABLE 2. The hyper-parameters of the CNN network.

Hyper Parameters	Value
Number of convolution layers	3
Number of pooling layers	1
Number of hidden layer neurons	2500
Output feature shape	$(N, 16, C)$
Activation function	ReLU
Drop probability	0.5

B. THE DISCRETE WAVELET TRANSFORMS

For feature extraction, methods such as fast Fourier transform (FFT) [29], discrete wavelet transforms (DWT) and common spatial patterns (CSP) are frequently employed. DWT is particularly popular for time–frequency sequence since it has the ability to achieve high resolutions in the time domain. The DWT can decompose sequence into its components in different frequency bands. This make the DWT perform well on spectral multiresolution and more applicable to EEG processing than FFT [30].

The DWT can decompose a signal into its components in different frequency bands. The DWT of a signal f (t) is given as follows:

$$W_{j,k}(f, g_{j,k}) = 2^{-\frac{1}{2}} \sum_{n=-\infty}^{+\infty} f(\mathrm{t})\bar{g}(2^{-j}t - k) \quad j, k \in Z \qquad (3)$$

where $g_{j,k}(t) = 2^{-\frac{j}{2}} g(2^{-j}t - k)$ is a wavelet sequence, j and k are the frequency resolution and time of the transform, respectively. In wavelet analysis, the low-frequency, high-scale component of the signal is approximated as L and the high-frequency, low-scale component as H. The extending scheme showing the components of a multi-level analysis

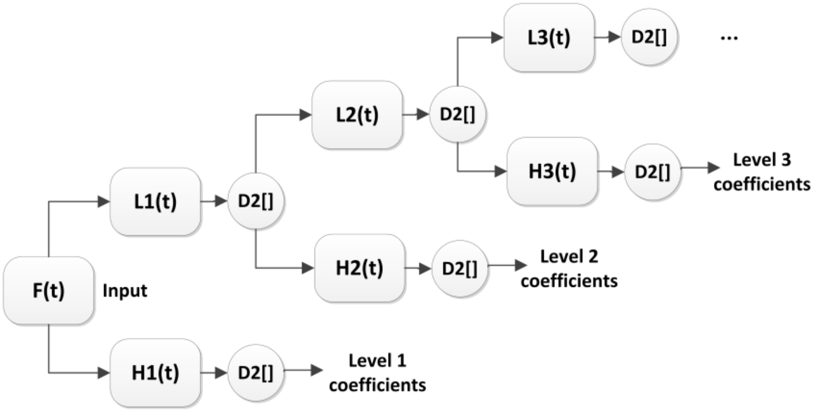

FIGURE 3. The hierarchically organized decomposition of DWT.

is depicted in Fig.3. $D_2[*]$ denotes the down sampling by a factor of 2 such as:

$$D_2[L_1(n)] = L_1(2n) \tag{4}$$

When the EEG is sampled with f_s, the corresponding bands can be denoted as follows:

$$\left[0, \frac{f_s}{2^{L+1}}\right], \left[\frac{f_s}{2^{L+1}}, \frac{f_s}{2^L}\right], \cdots, \left[\frac{f_s}{2^2}, \frac{f_s}{2}\right] \tag{5}$$

The temporal features extracted by DWT is maintained the spatial features which generate from the CNN. Then the LSTM recurrent networks classifier can make full use of these temporal features. The temporal features can be calculated in:

$$F_j = \sum_{i=1}^{G} S_{i,j} A_i \tag{6}$$

where $S_{i,j}A_i$ represents wavelet packet coefficients A_i work on i-th spatial feature S_i. G is the spatial feature number (G = 16) which generated from the CNN. The accumulation represents the fusion for serial features.

C. BATCH NORMALIZATION

In deep neural network, with the growing network depth, learning rate which control the speed of gradient descent in unnormalized networks is further limited by divergence due the magnitude of activations growing exponentially and gradient-information becomes less input-sensitive for unnormalized networks which limits possible learning rates [31], [32]. Hence, we employ batch normalization (BN) technique to improve generalization and accelerate training by normalizing inputs. For an L-dimensional input as mentioned above, we normalize each dimension as:

$$BN(X_i) = (X_i - E(X_i))/\sqrt{Var(X_i)} \tag{7}$$

$$E(X_i) = \frac{1}{m}\sum_{i=1}^{L} X_i \tag{8}$$

$$Var(X_i) = \frac{1}{m}\sum_{i=1}^{L} [X_i - E(X_i)]^2 \tag{9}$$

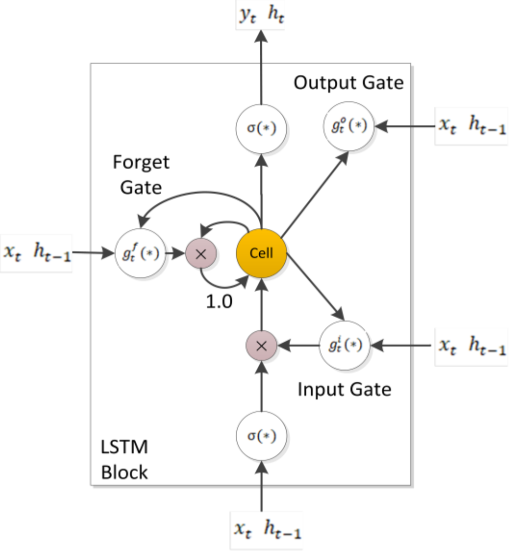

FIGURE 4. The LSTM unit block.

where X_i is the vector that needs to be normalized. $E(X_i)$ and $Var\ (X_i)$ are the expectation and variance of the current mini-batch of X_i, respectively. In our framework, we then use Batch Normalization to normalize the fusion feature to have a mean of 0 and standard deviation of 1. The batch size is set to correspond to our test data size. Then, the normalized features are introduced to our LSTM model.

D. THE LSTM NETWORK AND TRAINING PROCESS

In Fig.4, a typical LSTM unit block is composed of four main components: a cell, an input gate, an output gate and a forget gate. These units receive the activation sequence from different sources and control each cell's activation by the designed multipliers. The cell is designed to "remember" values over arbitrary time intervals, hence the word "memory" in LSTM. The LSTM networks can propagate errors and preserve signals much longer than traditional RNNs. Furthermore, Input and output gate function for setting input and output of the network while forget gate about setting the cell memory. The LSTM gates also can keep the rest network from adjusting the contents of the memory cells. Thus, gates play the important role in LSTM network due to responsibility for setting up memory cells in processing the stored information.

The LSTM used in this work is implemented by the following composite function:

$$g_t^c = g_t^f \otimes g_{t-1}^c + g_t^i \otimes \sigma(W_c x_t + U_c g_{t-1}^c + b_c) \tag{10}$$

$$g_t^f = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{11}$$

$$g_t^i = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{12}$$

$$g_t^o = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{13}$$

$$h_t = g_t^o \otimes \tanh(g_t^c) \tag{14}$$

where f, i, o and c denote the forget gate, input gate, output gate and cell state, respectively. W, U and b are the weight matrices and bias vector parameters which need to be learned during training. The σ denotes the element-wise sigmoid function, and $\otimes$ denotes the Hadamard product.

During the training, the gradient of the objective function with respect to each parameter can be calculated efficiently via back propagation over the whole network. Obtaining the back-propagation formulas is tedious, thus we list some indispensable details below to elaborate our work [33], [34]. For each memory block and error passed to the hidden vector δ_t^h the derivatives of each gate are computed as follow:

$$\delta_t^h = \frac{do}{dh_t} \tag{15}$$

$$\Delta_t^h = \delta_t^h \otimes \tanh(\delta_t^c) \otimes \theta'(g_t^o) \tag{16}$$

$$\Delta_t^f = \delta_t^c \otimes g_{t-1}^c \otimes \theta'(g_t^f) \tag{17}$$

$$\Delta_t^i = \delta_t^c \otimes \tanh(x_t) \otimes \theta'(g_t^i) \tag{18}$$

where $\theta'(*)$ is the element-wise derivative of the logistic function over vector $*$, and δ_t^c is the derivative of the cell vector.

The LSTM is theoretically powerful than alternative RNNs and other sequence learning methods due to its ability to learn from observations when it spends long time lags between relevant events. We used a sequential way to optimize the LSTM parameters. We firstly set the unit numbers to be 500 and optimized it. 200-time steps were chosen as it was long enough to capture the previous temporal correlation and not too complicated to calculate and training. Then we choose out the best weight though the training. The optimized parameter values for LSTM are detailed in Table 3. Final output unit produce a prediction at every time step. Given the softmax classifier, the predicted output of our architecture (expressed by conditional probabilities) [35] is denoted with the input X and label Y, defined as,

$$Y_j' = P(Y_i|f(X_j, w)) = \frac{\exp(f_i(X_j, w))}{\sum_{i=1}^{C}(f_i(X_j, w))} \tag{19}$$

through minimizing the sum of the cross-entropy losses [36], high probabilities will be assigned to the correct labels:

$$w = \arg\min \frac{1}{N}\sum_{j=1}^{N} Y_j \log(Y_j') + (1 - Y_j)\log(1 - Y_j') \tag{20}$$

TABLE 3. The hyper-parameters of the LSTM network.

Major Parameters	Value
Number of units	500
Cost function	cross-entropy
Learning rate	0.02
Classifier	softmax

IV. EXPERIMENTAL RESULTS

In this section, systematic and extensive experiments have been conducted on a public dataset and a local dataset to validate the performance of the proposed architecture for MI-EEG decoding. We provide comparative results of model training and spatial performance against other models. Moreover, the model feature patterns from each layer analysis are reported. Last, we evaluate our model's performance on training efficiency and overall classification comparison.

A. COMPARATIVE RESULTS

To validate the performance of our joint deep learning network for MI-EEG decoding, experimental comparisons are performed with other state-of-the-art methods, including Linear Discriminant Analysis (LDA), Naive Bayes (NB), support vector machine (SVM) and single LSTM. The traditional classifiers are implemented with python platform and Machine Learning Toolbox by applying the default parameters in MATLAB 2016a. CNN and LSTM are implemented using TensorFlow and Theano package in python platform. The results in terms of mean classification accuracy in the public datasets (D_2, D_3, D_4) are given in Table 4. The horizontal terms represent different subjects we selected from each dataset (three subjects in every dataset) as our comparison object.

As shown in Table 4, the proposed method achieved better performance than the other state-of-the-art methods for all the datasets. In these subjects, the accuracy has been increased in average over 3% compared with other methods. We can also note that the classification accuracy of the proposed method has performed significantly well in D_3 and D_4. To verify whether the performance of the joint model is statistically significant, we evaluate the CNN and LSTM models against the joint model, proving its effectiveness. In addition, we provide the comparison results within and without DWT which suggest this temporal feature extractor benefit the recognition. Finally, the mean classification accuracy for public datasets through main deep learning method is illustrated in bar chart. (see Fig. 5)

The classification accuracy of the experiments in private dataset is listed in Table 5 where the proposed method

TABLE 4. The recognition accuracy of different methods (public data).

Method	D_2			D_3			D_4		
	Sub$_A$	**Sub$_B$**	**Sub$_C$**	**Sub$_D$**	**Sub$_E$**	**Sub$_F$**	**Sub$_G$**	**Sub$_H$**	**Sub$_I$**
LDA	73.4	63.5	69.3	75.2	69.4	68.5	77.7	74.5	69.4
k-NN	61.3	65.2	68.5	66.9	68.7	72.5	67.9	68.9	70.3
NB	76.2	71.5	73.4	81.1	76.4	77.2	80.3	73.6	75.4
SVM	74.5	73.5	78.3	79.8	77.6	78.4	81.3	75.5	77.5
RF	75.9	79.2	77.0	81.2	78.5	77.4	83.1	78.3	79.6
CNN	77.3	78.4	77.3	83.9	82.0	80.6	79.3	80.2	82.7
LSTM	80.1	80.8	81.4	82.4	83.9	81.5	82.1	82.6	83.5
CNN-LSTM without DWT	80.4	81.5	79.6	83.3	84.4	83.9	84.5	83.2	83.7
CNN- LSTM with DWT	**84.3**	**86.2**	**85.3**	**87.3**	**89.3**	**88.5**	**88.3**	**90.2**	**87.9**

TABLE 5. The recognition accuracy of different methods (private data).

Method	Subject							Avg.
	Sub$_A$	**Sub$_B$**	**Sub$_C$**	**Sub$_D$**	**Sub$_E$**	**Sub$_F$**	**Sub$_G$**	
LDA	67.7	70.3	72.4	68.2	69.8	71.5	66.3	69.5
SVM	70.6	74.3	71.3	75.1	69.2	73.1	77.4	73.0
RF	75.8	77.2	78.3	79.4	73.4	78.3	69.3	75.9
CNN	**82.6**	79.3	78.8	79.4	75.5	83.5	78.3	79.5
LSTM	78.3	82.1	87.7	75.7	82.4	**90.3**	84.5	83.0
CNN-LSTM without DWT	81.3	81.6	88.2	79.6	83.4	84.2	83.2	85.2
CNN- LSTM with DWT	**82.6**	**84.4**	**91.6**	**83.8**	**91.2**	86.7	**88.3**	**86.7**

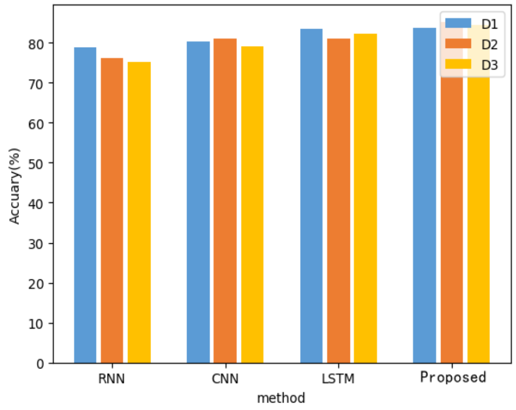

FIGURE 5. Comparison of accuracy among four methods with private data.

TABLE 6. The comparative results with different feature extractor.

Method	D_1	D_2	D_3	D_4
FFT	77.7	80.3	82.4	78.2
ICA	83.8	**85.8**	83.5	84.4
CSP	84.3	83.7	79.6	85.1
CNN+DWT	**86.7**	85.3	**88.3**	**88.8**

TABLE 7. The MI-EEG recognition comparison with other previous model (public data).

Feature extraction	Classification	Accuracy (%)
CSP [37]	SVM	75.93
CSP [38]	LDA	86.43
CNN [39]	SAE	86.41
FFT+DWT [40]	RBM	85.38
CNN+DWT	LSTM	**87.36**

outperforms the other methods for all the subjects except **sub$_F$**. The average classification accuracy has been increased about 4% compared with the second-best method. In addition, we can obviously learn that the DWT take effect both in public and private data. It can effectively improve the classification accuracy of MI-EEG.

We take comparison between the combination of CNN and DWT used in this work and aforementioned method of feature extraction. The comparison results of classification accuracy in Table 6 reveals that the ICA and CSP almost have the same performance and the combination of CNN and DWT outperform the other methods except in D_2 data set which suggest that the MI-EEG we chose from D_2 may involve less spatial or time-frequency dependence. As indicated in Table 7, the average accuracy of proposed method has an increased recognition rate at least about 1%. The stacked autoencoders (SAE) and restricted Boltzmann machine (RBM) are deep learning architecture and the remaining are machine learning method.

B. THE PARAMETER TUNING PERFORMANCE OF TRAINING PROCESS

Fig. 6 show that the proposed approach performs differently under different learning rates in each component. The final selected learning rates are 0.004 and 0.02 for CNN, and LSTM, respectively.

In order to observe the convergence process of our model at the parameter-tuning stage, the error rate curves of four datasets are showed in Fig. 7, which expose the percentage of the incorrectly classified samples in the training set. In our proposed scheme, the training process converges within about 100 epochs except subject A (over 200 epoch).

Fig. 8. presents the model performance when data used as fuel. The results illustrate that when near 70% of the

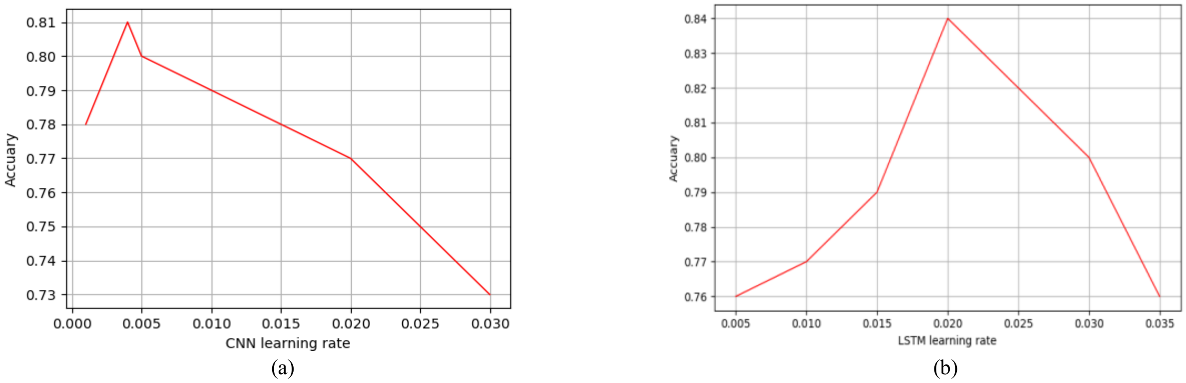

FIGURE 6. The learning rate comparison between RNN and LSTM. (a) CNN learning rate (b) LSTM learning.

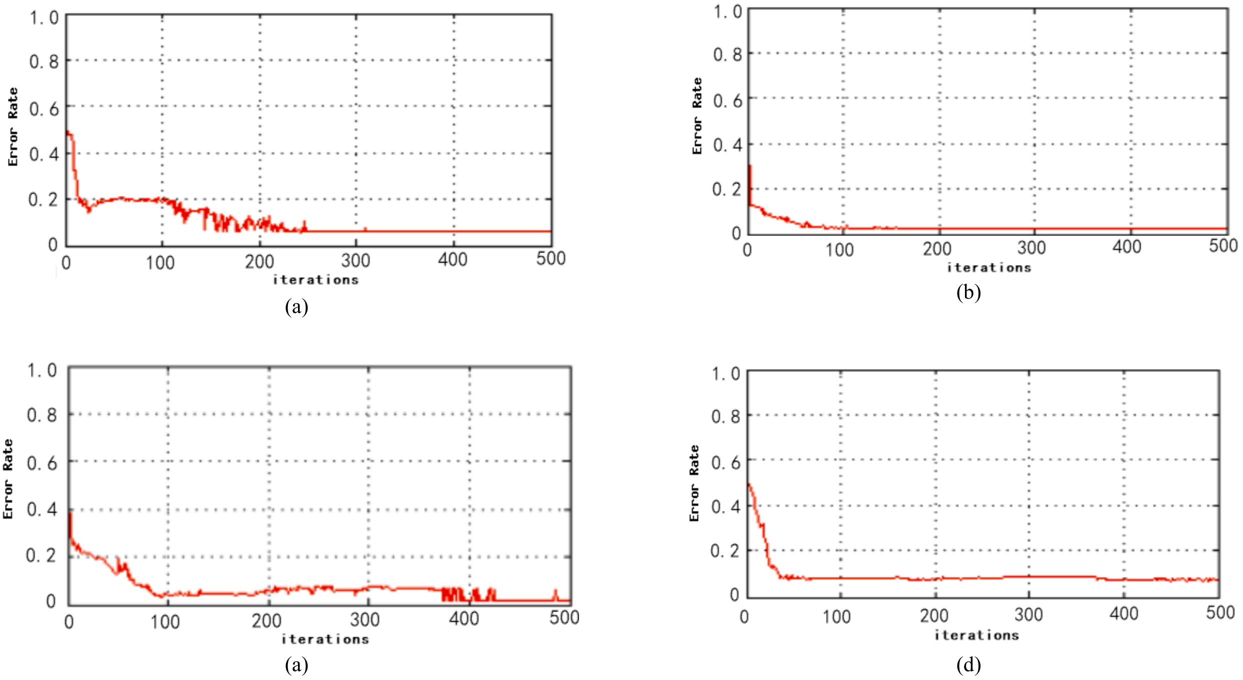

FIGURE 7. Error rate of training of different datasets (vertical term is the error rate while horizontal term presents the epochs of training). (a) D_1. (b) D_2. (c) D_3(d) D_4.

training set has been trained, our model recognition reaches a high accuracy of 89% and follow a slight fluctuation in accuracy with the remaining training data which reveal the tiny existence of overfitting. We can also learn from Fig. 8 that the training time changes nearly linearly with the scale of the training data.

From Table 8, LSTM take substantially longer to train than using batch normalization, especially the private data D1, which suggest that the gradient vanishing problem is effectively prevented by BN and it thus accelerate the convergence of training.

C. THE SPATIAL PERFORMANCE AND RECEIVER OPERATING CHARACTERISTIC CURVES OF OUR MODEL

A very important parameter was the number of electrodes in the electrode selection step of feature extraction. Too many electrodes can lead to over-fitting and estimation issues, while too few channels lead to a loss of information. Taking the spatial features which we filtered through the CNN into account, we can find out the most discriminant electrodes according to the weight, as shown in Table 9. This result approximately coheres with the MI-EEG data pre-processing part which we mentioned above. Not only the electrode selection but also high-level spatial feature presentations which is obtained through the spatial convolution help our recognition model to achieve high performance.

The receiver operating characteristic (ROC) curves of four methods for testing set are shown in Fig.9. For our private data, we define the left hand and right foot movement as positive class and negative class, respectively. According to Fig.10, the AUC (area under curve) of the ROC curve of proposed method is larger than those of the other three methods,

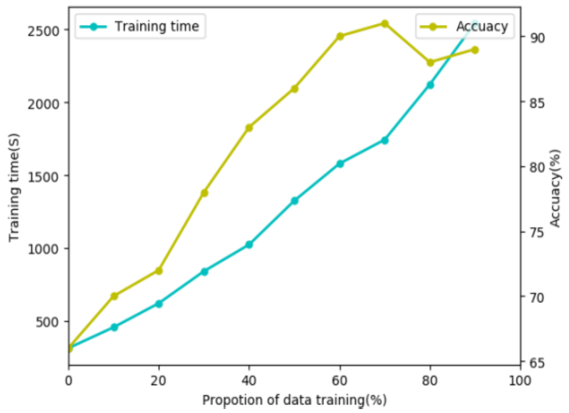

FIGURE 8. The performance analysis during data training process.

TABLE 8. Training times across subjects within and without BN.

	D_1	D_2	D_3	D_4
LSTM	00:46:27	00:28:33	00:37:52	00:27:32
BN-LSTM	00:34:53	00:20:49	00:29:10	00:19:52

TABLE 9. Most informative electrodes of subject A-D in D_1.

Data	Electrodes			
Sub$_{\mathrm{A}}$	C_3	P_4	P_z	O_2
	C_4	F_z	O_1	P_5
Sub$_{\mathrm{B}}$	C_4	C_3	P_3	P_4
	O_1	C_z	P_5	P_z
Sub$_{\mathrm{C}}$	C_4	P_3	P_2	P_5
	C_5	C_3	F_z	P_4
Sub$_{\mathrm{D}}$	C_4	P_3	O_1	P_4
	C_3	P_6	P_3	O_2

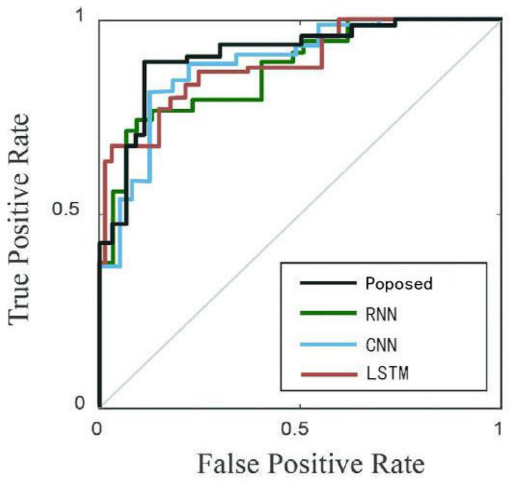

FIGURE 9. Comparison with ROC curves of four methods (D_1 dataset.

which suggest that the proposed model has a better capacity of discernment.

V. CONCLUSION AND FUTURE WORK

In this paper, a fusion feature extraction LSTM network was proposed for decoding raw EEG signals. The model employs CNN, DWT, BN and LSTM to learn the temporal and spatial dependency features from the input EEG raw data. The features are then fused to capture the temporal correlation and adaptive filtering is employed to incorporate temporal information into the system. We elaborate the CNN and LSTM capacity to learn high level EEG features consisted of low-level ones, after feature extraction by DWT. We evaluated our approach on a public and a private MI-EEG dataset with the results indicating that the proposed scheme is relatively robust to the BCIs and universally suitable for MI-EEG decoding. The results indicate that our proposed model can further improve classification performance compared to other methods. The application of combination of CNN and DWT in EEG analysis might possibly pave a way for classical spatio-temporal feature analysis in bioelectric signal. LSTM network can be promoted also for EEG decoding and recognition, and for exploring more complex EEG features.

Future work will attempt to introduce the attention region mechanism in order to extract more interesting information to improve the performance of the motor imagery BCIs. In addition, we will further extend the proposed MI-based BCI recognition model for more potential input data, such as near-infrared spectroscopy (NIRS) and functional magnetic resonance imaging (fMRI).

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their constructive comments. Furthermore, the maintenance and development of theno and keras on Github made the implementations of the algorithms much easier and faster.

REFERENCES

[1] Z. Emami and T. Chau, "Investigating the effects of visual distractors on the performance of a motor imagery brain-computer interface," *Clin. Neurophysiol.*, vol. 129, no. 6, pp. 1268–1275, 2018.

[2] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi, "A review of classification algorithms for EEG-based brain–computer interfaces," *J. Neural Eng.*, vol. 4, no. 2, p. R1, 2007.

[3] A. M. Chiarelli, P. Croce, A. Merla, and F. Zappasodi, "Deep learning for hybrid EEG-fNIRS brain–computer interface: Application to motor imagery classification," *J. Neural Eng.*, vol. 15, no. 3, p. 036028, Jun. 2018, doi: 10.1088.

[4] M.-P. Hosseini, D. Pompili, K. Elisevich, and H. Soltanian-Zadeh, "Optimized deep learning for EEG big data and seizure prediction BCI via Internet of Things," *IEEE Trans. Big Data*, vol. 3, no. 4, pp. 392–404, 2017.

[5] M.-A. Li, H.-N. Liu, W. Zhu, and J.-F. Yang, "Applying improved multiscale fuzzy entropy for feature extraction of MI-EEG," *Appl. Sci.*, vol. 7, no. 1, p. 92, 2017.

[6] Z. Tang, C. Li, and S. Sun, "Single-trial EEG classification of motor imagery using deep convolutional neural networks," *Int. J. Light Electron Opt.*, vol. 130, pp. 11–18, Oct. 2016.

[7] H. B. Kappes and C. K. Morewedge, "Mental simulation as substitute for experience," *Soc. Pers. Psychol. Compass*, vol. 10, no. 7, pp. 405–420, 2016.

[8] M. Tariq, L. Uhlenberg, and P. Trivailo, "Mu-beta rhythm ERD/ERS quantification for foot motor execution and imagery tasks in BCI applications," in *Proc. IEEE Int. Conf. Cognit. Infocommun.*, vol. 13, no. 6, Sep. 2017, pp. 91–96, doi: 10.1109/CogInfoCom.2017.8268222.

[9] Y.-H. Liu, L.-F. Lin, C.-W. Chou, Y. Chang, Y.-T. Hsiao, and W.-C. Hsu, "Analysis of electroencephalography event-related desynchronisation and synchronisation induced by lower-limb stepping motor imagery," *J. Med. Biol. Eng.*, Mar. 2018, doi: 10.1007.

[10] X. Zhang, L. Yao, Q. Z. Sheng, S. S. Kanhere, T. Gu, and D. Zhang. (2017). "Converting your thoughts to texts: Enabling brain typing via deep feature learning of EEG signals." [Online]. Available: https://arxiv.org/abs/1709.08820

[11] R. Masoomi and A. Khadem, "Enhancing LDA-based discrimination of left and right hand motor imagery: Outperforming the winner of BCI competition II," in *Proc. 2nd Int. Conf. Knowl.-Based Eng. Innov.*, Nov. 2015, pp. 392–398, doi: 10.1109/KBEI.2015.7436077.

[12] S. C. Yeh, M. Norrima, and P. Girijesh, "Automated classification and removal of EEG artifacts with SVM and wavelet-ICA," *IEEE J. Biomed. Health Inform.*, vol. 22, no. 3, pp. 664–670, May 2018.

[13] M. Miao, H. Zeng, A. Wang, C. Zhao, and F. Liu, "Discriminative spatial-frequency-temporal feature extraction and classification of motor imagery EEG: An sparse regression and Weighted Naïve Bayesian classifier-based approach," *J. Neurosci. Methods*, vol. 278, pp. 13–24, Feb. 2017.

[14] A. Turnip, A. I. Simbolon, M. F. Amri, P. Sihombing, R. H. Setiadi, and E. Mulyana, "Backpropagation neural networks training for EEG-SSVEP classification of emotion recognition," *Internetworking Indonesian J.*, vol. 9, no. 1, pp. 53–57, 2017.

[15] S. Sakhavi, C. Guan, and S. Yan, "Learning temporal information for brain-computer interface using convolutional neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5619–5629, Nov. 2018.

[16] R. Manor and A. B. Geva, "Convolutional neural network for multi-category rapid serial visual presentation BCI," *Front. Comput. Neurosci.*, vol. 9, pp. 1–12, Dec. 2015.

[17] N. K. Verma, L. S. V. S. Rao, and S. K. Sharma, "Motor imagery EEG signal classification on DWT and crosscorrelated signal features," in *Proc. 9th Int. Conf. Ind. Inf. Syst. (ICIIS)*, Dec. 2015, pp. 1–6.

[18] M. A. Naderi, "Analysis and classification of EEG signals using spectral analysis and recurrent neural networks," in *Proc. 17th Iranian Conf. Biomed. Eng.*, Nov. 2010, pp. 1–4.

[19] E. D. Übeyli, "Analysis of EEG signals by implementing eigenvector methods/recurrent neural networks," *Digit. Signal Process.*, vol. 19, no. 1, pp. 134–143, Jan. 2009.

[20] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee. (Mar. 2018). "Recent advances in recurrent neural networks." [Online]. Available: https://arxiv.org/abs/1801.01078

[21] J. Zhou, M. Meng, Y. Gao, Y. Ma, and Q. Zhang, "Classification of motor imagery EEG using wavelet envelope analysis and LSTM networks," in *Proc. Chin. Control Decis. Conf.*, Jul. 2018, pp. 5600–5605.

[22] M. Li, M. Zhang, X. Luo, and J. Yang, "Combined long short-term memory based network employing wavelet coefficients for MI-EEG recognition," in *Proc. IEEE Int. Conf. Mechatronics Automat.*, Aug. 2016, pp. 1971–1976.

[23] X. Bai, X. Wang, S. Zheng, and M. Yu, "The offline feature extraction of four-class motor imagery EEG based on ICA and wavelet-CSP," in *Proc. Chin. Control Conf.*, Sep. 2014, pp. 7189–7194.

[24] Y. Wei, Y. Jun, S. Lin, and L. Hong, "Improving classification accuracy using fuzzy method for BCI signals," *Bio-Med. Mater. Eng.*, vol. 24, no. 6, pp. 2937–2943, 2014.

[25] A. Koçanaoğullai, F. Quivira, and D. Erdoğmuş, "Incorporating temporal dependency on ERP based BCI," in *Proc. Int. Symp. Biomed. Imag.*, Apr. 2018, pp. 752–756.

[26] M. Liu, W. Wu, Z. Gu, Z. Yu, F. Qi, and Y. Li, "Deep learning based on batch normalization for P300 signal detection," *Neurocomputing*, vol. 275, pp. 288–297, Jan. 2018.

[27] A. T. Sözer and C. B. Fidan, "Novel spatial filter for SSVEP-based BCI: A generated reference filter approach," *Comput. Biol. Med.*, vol. 96, pp. 98–105, Feb. 2018.

[28] S.-M. Park, J.-Y. Kim, and K.-B. Sim, "EEG electrode selection method based on BPSO with channel impact factor for acquisition of significant brain signal," *Optik*, vol. 155, pp. 89–96, Feb. 2018.

[29] J. Lin and W. Wu, "An FPGA-based BCI system with SSVEP and phased coding techniques," *J. Technol.*, vol. 33, no. 1, pp. 53–62, 2018.

[30] A. A. Azamimi, S. A. Rahim, and A. Ibrahim, "Development of EEG-based epileptic detection using artificial neural network," in *Proc. Int. Symp. Elect. Comput. Eng.*, Jan. 2012, doi: 10.1109.

[31] M. Liu, W. Wu, Z. Gu, Z. Yu, F. F. Qi, and Y. Li, "Deep learning based on Batch Normalization for P300 signal detection," *Neurocomputing*, vol. 275, pp. 288–297, Jan. 2018.

[32] J. Bjorck, C. Gomes, and B. Selman. (2018). "Understanding batch normalization." [Online]. Available: https://arxiv.org/abs/1806.02375

[33] J. Thomas, T. Maszczyk, N. Sinha, T. Kluge, and J. Dauwels, "Deep learning-based classification for brain-computer interfaces," in *Proc. IEEE Int. Conf. Syst.*, Oct. 2017, pp. 234–239.

[34] Z. Xie, O. Schwartz, and A. Prasad, "Decoding of finger trajectory from ECoG using deep learning," *J. Neural Eng.*, vol. 15, no. 3, p. 036009, 2018.

[35] R. T. Schirrmeister *et al.*, "Deep learning with convolutional neural networks for EEG decoding and visualization," *Hum. Brain Mapping*, vol. 38, no. 11, pp. 5391–5420, 2017.

[36] A. M. Leite da Silva, R. A. González-Fernández, and C. Singh, "Generating capacity reliability evaluation based on monte carlo simulation and cross-entropy methods," *IEEE Trans. Power Syst.*, vol. 25, no. 1, pp. 129–137, Feb. 2010.

[37] L. Duan, Z. Hongxin, M. S. Khan, and M. Fang, "Recognition of motor imagery tasks for BCI using CSP and chaotic PSO twin SVM," *J. China Univ. Posts Telecommun.*, vol. 24, pp. 83–90, Jun. 2017.

[38] Y. Wang, S. Gao, and X. Gao, "Common spatial pattern method for channel selelction in motor imagery based brain–computer interface," in *Proc. 27th Annu. Conf. IEEE Eng. Med. Biol.*, Jan. 2006, pp. 5392–5395.

[39] Y. R. Tabar and U. Halici, "A novel deep learning approach for classification of EEG motor imagery signals," *J. Neural Eng.*, vol. 14, no. 1, p. 016003, 2016, doi: 10.1088

[40] N. Lu, T. Li, X. Ren, and H. Miao, "A deep learning scheme for motor imagery classification based on restricted Boltzmann machines," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 6, pp. 566–576, Jun. 2017.

JUN YANG received the M.S. degree in communication and information system from Yunnan University, Kunming, China, where he is currently pursuing the Ph.D. degree. He is also a Lecturer with the School of Information Engineering and Automation, Kunming Science and Technology University, China. He has authored several conference and journal papers on his research topic. His research includes deep learning and its application to brain–computer interface.

SHAOWEN YAO received the B.S. and M.S. degrees in telecommunication engineering from Yunnan University, China, in 1988 and 1991, respectively, and the Ph.D. degree in computer application technology from the University of Electronic Science and Technology of China in 2002. He is currently a Professor with the School of Software, Yunnan University. His current research interests include neural network theory and applications, cloud computing, and big data.

JIN WANG received the Ph.D. degree in computer science and engineering from Yuan Ze University, Taoyuan, Taiwan, and the Ph.D. degree in communication and information systems from Yunnan University, Kunming, China. He is currently a Lecturer with the School of Information Science and Engineering, Yunnan University. His research interests include natural language processing, text mining, and machine learning.